CLEAN VERSION OF PENDING CLAIMS

1.(Amended) A system comprising:

a bus;

a resource coupled to the bus; and

a plurality of entities coupled to the bus, at least one entity among the plurality of entities including a memory, wherein at least a portion of the memory of the at least one entity is selectively reset when the at least one entity has access to the resource, wherein the memory portion is reset if the entity does not currently have control of the resource, and wherein the memory portion is not reset when the at least one entity currently has control of the resource.

2. The system of claim 1, wherein the at least one entity is an integrated circuit.

3. The system of claim 1, wherein the resource includes memory.

4. The system of claim 1, further comprising a manager to manage at least one request from the plurality of entities to access the resource.

5. The system of claim 1, further comprising an arbiter coupled to the plurality of entities to arbitrate at least one bus request from the plurality of entities.

8.(Amended) An integrated circuit for allowing a resource to be controlled by a plurality of processors including a first and second processor, wherein the first processor includes a fast memory, the integrated circuit comprising:

a bus;

a central computing unit coupled to the bus; and

a switch mechanism, coupled to the central computing unit, to switch the control of the resource, wherein a portion of the fast memory of the first processor is reset when the control of the resource is switched to the second processor and not reset when control of the resource

remains with the first processor.

11. The integrated circuit of claim 8, wherein the switch mechanism is a hardware device.

12. The integrated circuit of claim 8, wherein the switch mechanism is a software switch.

13. The integrated circuit of claim 12, wherein the software switch is a Djikstra primitive.

14. The integrated circuit of claim 8, wherein the at least one resource is a hardware resource.

15. The integrated circuit of claim 14, wherein the hardware resource is a memory.

16. The integrated circuit of claim 8, further comprising a communications channel controller coupled to the bus.

17. The integrated circuit of claim 8, wherein the at least one resource is a software resource.

18. The integrated circuit of claim 17, wherein the software resource is a data structure.

19. The integrated circuit of claim 8, wherein the fast memory is cache memory.

20.(Amended) An integrated circuit for allowing a resource to be shared by a plurality of processors, including a first and second processor, wherein the first processor includes a fast memory, the integrated circuit comprising:

a bus;

a central computing unit coupled to the bus; and

a lock coupled to the central computing unit to reserve exclusive control of the resource, wherein a portion of the fast memory of the first processor is reset when the second processor obtains exclusive control of the resource from the first processor and not reset when exclusive

control remains with the first processor.

23. The integrated circuit of claim 20, wherein the lock is a hardware register.

24. The integrated circuit of claim 20, wherein the lock is a software semaphore.

25. The integrated circuit of claim 24, wherein the software semaphore is a binary semaphore.

26. The integrated circuit of claim 20, further comprising a communications channel controller coupled to the bus.

27. The integrated circuit of claim 20, wherein the fast memory is cache memory.

28. The integrated circuit of claim 27, wherein the cache memory is primary cache memory.

29. The integrated circuit of claim 27, wherein the cache memory is secondary cache memory.

30. (Amended) A data structure in a machine-readable medium for allowing a resource to be shared among a plurality of processors, at least one processor of the plurality of processors including a fast memory, the data structure comprising:

a state for indicating that the resource is under control; and

a first identifier for identifying a past processor that had exclusive control of the resource.

31. The data structure of claim 30, wherein the data structure is a class, the data structure further comprising an act for resetting at least a portion of the fast memory of the present processor.

32. The data structure of claim 31, further comprising a second identifier for identifying the present processor that has exclusive control of the resource.

33. The data structure of claim 32, further comprising an act for comparing the first identifier and the second identifier, and wherein the act for resetting the fast memory of the present processor being executed when the first identifier is different from the second identifier.

34. The data structure of claim 30, wherein the fast memory is cache memory.

35.(Amended) The data structure of claim 30, further comprising a data type that is adapted to represent at least one portion of the resource, wherein the data type includes at least one location of the at least one portion of the resource and at least one dimension of the at least one portion of the resource.

36.(Amended) The data structure of claim 30, further comprising a list that includes at least one location of at least one portion of the resource and at least one dimension of at least one portion of the resource.

37.(Amended) A method for allowing at least one resource to be shared among a plurality of processors, the method comprising:

obtaining exclusive control over the at least one resource by a present processor, the present processor including a fast memory;

identifying a past processor to obtain a first identity, wherein the past processor had exclusive control over the at least one resource;

identifying a present processor to obtain a second identity, the present processor having exclusive control over the at least one resource;

comparing the first identity and the second identity so as to determine if the present processor is different from the past processor; and

resetting a portion of the fast memory of the present processor when the past processor is

different from the present processor and not resetting a portion of the fast memory of the present processor when the past processor is the same as the present processor.

38. The method of claim 37, wherein identifying the present processor further comprises the fast memory as cache memory.

41. The method of claim 37, wherein the progression of the method is in the order presented.

42.(Amended) A method for scheduling access to a resource from among a plurality of processors, the method comprising:

obtaining access to the resource by a requesting processor, the requesting processor including a cache memory;

excluding access to the resource from the plurality of processors except for the requesting processor; and

resetting a portion of the cache memory of the requesting processor when the requesting processor is different from a processor that previously had access to the resource, and not resetting a portion of the cache memory of the requesting processor when the requesting processor is the same as a processor that previously had access to the resource.

43.(Amended) An integrated circuit for allowing a resource to be controlled by a plurality of processors, including a first and second processor, wherein the first processor includes a fast memory, the integrated circuit comprising:

a bus;

a central computing unit coupled to the bus;

a switch mechanism for switching the control of the resource; and

a lock, in a cooperative relationship with the switching mechanism, for reserving exclusive control of the resource to the first processor, wherein at least a portion of the fast memory of the first processor is reset when the second processor obtains exclusive control of the resource and not reset when the exclusive control of the resource remains with the first processor.

44. The integrated circuit of claim 43, wherein the fast memory is cache memory.

45. The integrated circuit of claim 43, further comprising a communications channel controller coupled to the bus, wherein the communications channel controller is receptive to diverse communications protocols.

46. The integrated circuit of claim 43, wherein the cooperative relationship of the switch mechanism and the lock maintains cache coherency.

49.(Amended) An integrated circuit for allowing a resource to be controlled by a plurality of processors, including a first and second processor, wherein the first processor includes a fast memory, the integrated circuit comprising:

a bus;

a central computing unit coupled to the bus; and

a scheduler, coupled to the central computing unit, for scheduling the control of the resource, wherein a portion of the fast memory of the first processor is reset when the resource becomes under the control of the second processor and not reset when the resource remains under the control of the first processor.

52.(Amended) A system comprising:

a bus;

a resource coupled to the bus;

a plurality of processors coupled to the bus including a first and second processor, wherein the first processor includes a fast memory; and

a switch mechanism, coupled to the bus, to switch the control of the resource, wherein a portion of the fast memory of the first processor is reset when the control of the resource is switched to the second processor and not reset when control of the resource remains with the first processor.

55. The system of claim 52, wherein the switch mechanism is a hardware device.

56. The system of claim 52, wherein the switch mechanism is a software switch.

57. The system of claim 56, wherein the software switch is a Djikstra primitive.

58. The system of claim 52, wherein the at least one resource is a hardware resource.

59. The system of claim 58, wherein the hardware resource is a memory.

60. The system of claim 52, wherein the at least one processor includes a communications channel controller.

61. The system of claim 52, wherein the at least one resource is a software resource.

62. The system of claim 61, wherein the software resource is a data structure.

63. The system of claim 52, wherein the fast memory is cache memory.

64.(Amended) A system comprising:

a bus;

a resource coupled to the bus;

a plurality of processors coupled to the bus including a first and second processor, wherein the first processor includes a fast memory; and

a lock to reserve exclusive control of the resource, wherein a portion of the fast memory of the first processor is reset when the second processor of the plurality of processors obtains exclusive control of the resource from the first processor and not reset when exclusive control of the resource remains with the first processor.

67. The system of claim 64, wherein the lock is a hardware register.

68. The system of claim 64, wherein the lock is a software semaphore.

69. The system of claim 68, wherein the software semaphore is a binary semaphore.

70. The system of claim 64, further comprising a communications channel controller coupled to the bus.

71. The system of claim 64, wherein the fast memory is cache memory.

72. The system of claim 71, wherein the cache memory is primary cache memory.

73. The system of claim 71, wherein the cache memory is secondary cache memory.

74.(Amended) A system comprising:

a bus;

a resource coupled to the bus;

a plurality of processors coupled to the bus including a first and second processor, wherein the first processor includes a fast memory;

a switch mechanism to switch the control of the resource; and

a lock, in a cooperative relationship with the switching mechanism, to reserve exclusive control of the resource to a first processor, wherein at least a portion of the fast memory of the first processor is reset when the second processor obtains exclusive control of the resource and not reset when the exclusive control of the resource remains with the first processor.

75. The system of claim 74, wherein the fast memory is cache memory.

76. The system of claim 74, wherein the at least one processor includes a communications channel controller, wherein the communications channel controller is receptive to diverse communications protocols.

77. The system of claim 74, wherein the cooperative relationship of the switch mechanism and the lock maintains cache coherency.